Substitute Specification – Marked-Up Copy

TITLE OF THE INVENTION:
Network Server and Method of Discovery of a Network Node

~~CROSS-REFERENCE TO RELATED APPLICATIONS:~~
~~Not Applicable~~

~~STATEMENT REGARDING FEDERALLY FUNDED RESEARCH OR DEVELOPMENT:~~
~~Not Applicable~~

~~INCORPORATION BY REFERENCE OF MATERIAL SUBMITTED ON COMPACT DISC:~~
~~Not Applicable~~

~~BACKGROUND OF THE INVENTION:~~

Related Applications

[001] The present application is based on, and claims priority from, German Application Number 103 38 113.9, filed August 15, 2003, the disclosure of which is hereby incorporated by reference herein in its entirety.

Technical Field

[002] This invention relates to computer systems and more particularly to computer networks that interconnect network nodes, such as computers and computer peripherals. Even more particularly, the invention relates to discovery of network nodes.

Background Art

[003] Network Management Systems like the OpenView Network Node Manager product are designed to discover network topology (i.e., a list of all network node in a domain, their type, and their connections), monitor the health of each network node, and report problems to the network administrator. OpenView Network Node Manager (NNM) is a product distributed by Hewlett-Packard Company of Palo Alto, Calif.

[004] The monitoring function of such a system is usually performed by a specialized computer program which periodically polls each network element and gathers data which is indicative of the network element's health. A monitor program typically runs on a single host.

However, in distributed networks, monitors may run on various nodes in the network, with each monitor reporting its results to a centralized display.

[005] Network discovery is performed periodically using Internet control message protocol (ICMP) polling (pings and mask requests), simple network management protocol (SNMP) polling and / or other diagnostic requests. In addition or as an alternative to these protocols Desktop Management Interface (DMI) or Windows Management Instrumentation (WMI) which is also referred to as WBEM can be used.

[006] For example by means of SNMP polling a configuration check operation is performed on each node. By default this configuration check operation is performed once each day whereby this period is configurable. This way it is determined at periodic intervals if any relevant configuration information has changed since the last check. More technical detail of the internals of the NNM product is disclosed on (http://support.openview.hp.com/pdf/dev_ov_netmon96.pdf), the entirety of which is herein incorporated by reference.

[007] Elements of the code for discovering the topology of a plurality of network elements and the code for periodically polling a plurality of network interfaces associated with the plurality of network elements are also disclosed in U.S. Pat. No. 5,185,860 of Wu entitled "Automatic Discovery of Network Elements, and in U.S. Pat. No. 5,276,789 of Besaw et al. entitled "Graphic Display of Network Topology". Both of these patents are hereby incorporated by reference for all that they disclose.

[008] A disadvantage of scheduled discovery which is performed by the NNM product is that the discovery procedure can negatively affect network performance when the discovery procedure is carried out. This is why the discovery schedule is typically set such that the discovery procedure is carried out when the network load is low, i.e. during the night. However, performing the discovery at predetermined time intervals has the disadvantage that network nodes which are connected to the network only temporarily can be missed by the discovery. In particular this applies to portable computers, such as lap top computers which are frequently connected and disconnected to the network by means of a docking station.

Summary of the Invention

[009] The present invention provides a new and improved method of and apparatus for discovering a network node. Initially an access request is received from a network node. In response to the access request a discovery request is generated and transmitted to a discovery server. The discovery server performs a discovery procedure for the network node from which the access request has been received.

[0010] In comparison to a scheduled discovery the method and apparatus set forth in the immediately preceding paragraph has the advantage that network nodes, which are only temporarily connected to the network, are always discovered while making minimal usage of network bandwidth resources for the discovery procedure as not the whole network but only the network node identified by the access request is discovered. Another advantage is that no extra software needs to be installed on the network nodes as the discovery procedure can be initiated by a standard access request, such as a log on request or an Internet protocol (IP) address request.

[0011] A further advantage is that the unplanned discovery of network nodes can be performed centrally by one discovery server maintaining the discovery rules and collecting the discovery results or a few interconnected discovery servers. This greatly simplifies the network administration task.

[0012] In accordance with a preferred embodiment of the invention a discovery procedure is only performed for a given network node after a minimum amount of time has lapsed since the last discovery procedure had been performed. This way unnecessary discovery procedures are avoided when a network node is frequently disconnected and reconnected to the computer network. For example, the configuration of a lap top computer usually does not change very often. However, a lap top computer system can be removed and reconnected to the computer network within short time intervals, especially if a docking station is used.

[0013] In accordance with a further preferred embodiment of the invention the discovery procedure is initiated by a log on request received by a network server, such as a domain controller, from a network node. Typically the user of the network node needs to enter his or her user ID and password for authentication. After the log on procedure has been completed

successfully a discovery request is generated and sent to a discovery server from the network server.

[0014] In accordance with a further preferred embodiment of the invention the network node request which initiates the discovery procedure is an IP address request. For example, when a computer peripheral, such as a printer, is connected to the computer network, the computer peripheral sends an IP address request which is processed by ~~a~~ Dynamic Host Configuration Protocol (DHCP) server. The DHCP server assigns an IP address to the requesting network node. After the IP address has been assigned to the network node a discovery request is generated and transmitted to the discovery server together with the IP address.

Brief Description of the Drawings

[0015] In the following preferred embodiments of the invention will be described, by way of example, and with reference to the drawings in which:

[0016] Figure 1 is illustrative of a block diagram of a computer network having a network server being coupled to a discovery server,

[0017] Figure 2 is illustrative of a flow chart for performing a method of discovery of a network node,

[0018] Figure 3 is a block diagram of a computer network having a temporarily connected portable computer,

[0019] Figure 4 is illustrative of a block diagram of a computer network having a DHCP server,

[0020] Figure 5 is illustrative of a flow chart of an alternative method of discovery of the network node.

Detailed Description of the Drawing

[0021] Figure 1 includes a network server 100 of a computer network having a plurality of network nodes. For ease of explanation only network node 102 of the computer network is shown in Figure 1. Network node 102 is not permanently coupled to the computer network but is

disconnected and reconnected from time to time. Network node 102 has an assigned IP address for addressing of network node 102 by means of the transmission control protocol / Internet protocol (TCP / IP).

[0022] Network server 100 has program component 104 for processing of access request 106 received from network node 102. For example program component 104 serves for authentication purposes of network node 102 and / or of a user of network node 102.

[0023] Further network server 100 has program component 108 which serves for initiation of a discovery procedure. After successful completion of the processing of access request 106 by program component 104, program component 108 generates discovery request 110 and sends discovery request 110 together with the IP address of network node 102 to discovery server 112.

[0024] Discovery server 112 has memory 114 for storing of the IP address received with discovery request 110. For example memory 114 is a FIFO buffer for storing a stack of IP addresses.

[0025] Discovery server 112 has discovery program 116 for performing a discovery procedure for network node 102. The discovery procedure is performed by ICMP and / or SNMP polls 118 in order to determine the network topology, i.e. other nodes to which network node 102 is connected, to discover configuration information of network node 102 and / or other status information of network node 102.

[0026] In operation, when network node 102 is physically connected to the network, it sends access request 106 to network server 100 in order to obtain access to the network resources. When access request 106 is received by network server 100 this invokes program component 104. Program component 104 processes access request 106. For example, program component 104 performs an authentication procedure prior to granting access to the network resources. After successful completion of the processing of access request 106 by program component 104 access is granted to network node 102 and program component 108 is invoked by program component 104.

[0027] Program component 108 generates discovery request 110 which comprises the IP address of network node 102. It is to be noted that in the embodiment considered here fixed IP addresses are being used, i.e. the IP address of network node 102 is assigned once and then fixed.

[0028] Program component 108 sends discovery request 110 with the IP address to discovery server 112. Discovery server 112 stores the IP address received with discovery request 110 in memory 114. Further discovery program 116 is invoked for performing a discovery procedure for network node 102 which is identified by its IP address. This is performed by ICMP and / or SNMP polls.

[0029] Discovery server 112 can receive a sequence of discovery requests 110 containing various IP addresses of various network nodes which have requested access to the computer network. This sequence of IP addresses is stored as a stack in memory 114. The stack of IP addresses is processed by discovery program 116 in first-in-first-out order.

[0030] It is to be noted that network server 100 and discovery server 112 are implemented on different server computers; alternatively network server 100 and discovery server 112 are implemented on the same server computer which is partitioned correspondingly. In particular the discovery server 112 can be implemented by any server computer running the relevant program, such as the network server computer 100.

[0031] Figure 2 is a corresponding flowchart. In step 200 an access request is received from a removable network node which has been physically reconnected to the network. The access request from the network node is directed towards logical re-coupling of the network node to the computer network, i.e. the granting of access to network resources.

[0032] In step 202 the access request is processed. For example prior to granting of access the authenticity of the network node 102 and/or of its user and / or of its access rights are checked. After successful authentication and / or successful check of the access rights the network node is granted access to the network and is logically reconnected.

[0033] ~~In~~ As indicated by step 204, in response to the logical reconnection of the network node a discovery request 110 is generated by server 100 and sent to the discovery server 112. The discovery request contains an identifier of the network node, such as its node ID. Preferably the IP address of the network node is used as a node ID.

[0034] Alternatively the name or other addressing information of the node can be used as an identifier. In response to the discovery ~~request~~ request, the discovery server 112 performs a discovery procedure for the network node identified by the node ID in the discovery request in step 206.

[0035] The trigger for performing the discovery procedure of step 206 is the discovery request and not a prescheduled point of time. This means that a discovery procedure is only initiated if the network node is actually coupled to the network.

[0036] Figure 3 is an alternative embodiment based on the embodiment of Figure 1. Like elements of the embodiments of Figures 1 and 3 are designated by like reference numerals having added 200 to the reference numerals in the embodiment of Figure 3.

[0037] In the embodiment shown in Figure 3 network node 302 comprises portable computer 320 which is physically coupled to the computer network through docking station 322. For example portable computer 320 is a lap top computer, palm top computer or other mobile computing device.

[0038] In operation portable computer 320 is reconnected to the computer network by inserting portable computer 320 into docking station 322 and booting portable computer 320. Next a log on dialogue is started between network server 300 and portable computer 320.

[0039] If a Microsoft windows operating system is used network server 300 plays the role of a domain controller for portable computer 320. Log on request 306 is processed by program component 304 of network server 300. When the user of portable computer 320 has correctly entered a valid user ID and ~~password~~ password, portable computer 320 is logged on and is thus logically reconnected to the network.

[0040] As a ~~consequence~~ consequence, program component 304 invokes program component 308 for initiation of the discovery procedure for portable computer 320. The IP address of portable computer 320 which is known to network server 300 is transmitted as part of discovery request 310 to discovery server 312 which performs a discovery procedure analogous to the discovery procedure as explained above with respect to Figures 1 and 2.

[0041] Figure 4 is a block diagram of a further preferred embodiment where like elements are referenced by like reference numerals as in Figure 1 having added 300 to the reference numerals.

[0042] In the embodiment shown in Figure 4 network server 400 is a DHCP server. DHCP provides automated configuration services including automated assignment of IP addresses. Network server 400 has a corresponding DHCP services program component 404.

[0043] In the preferred embodiment considered here, network node 402 is a computer peripheral device, such as a printer. Computer peripheral 402 has a media access control (MAC) address which is stored in permanent storage 420 of computer peripheral 402.

[0044] Network server 400 has time stamp program component 422. Time stamp program component 422 serves to provide a time stamp for discovery request 410 which is generated by program component 408 for discovery initiation.

[0045] In operation when computer peripheral 402 is physically coupled to the computer network an IP address request 406 is automatically generated by computer peripheral 402 based on the MAC address stored in storage 420 in accordance with the DHCP protocol. In response to IP address request 406 network server 400 invokes DHCP services program component 404 for processing of IP address request 406.

[0046] DHCP services program component 404 assigns an IP address to computer peripheral 402. Next program component 408 is invoked which generates discovery request 410 which includes the newly assigned IP address of computer peripheral 402. Discovery request 410 is time stamped by time stamp program component 422 and then sent from network server 400 to discovery server 412.

[0047] Discovery server 412 has a discovery log file 424 in which the IP addresses and time stamps of previously received discovery requests for which discovery procedures have been performed are stored. For example discovery log 424 has a tabular structure, e.g. for each request a request number, time stamp and IP address are stored in the discovery log 424. Discovery server 412 queries discovery log 424 using the IP address of discovery request 410 as a key. If an entry in discovery log 424 with that IP address is identified the corresponding time stamp stored in the discovery log 424 and the time stamp of discovery request 410 are ~~corapared.~~ compared.

[0048] If time stamp of discovery request 410 is a predetermined amount of time later then the time stamp stored in discovery log 424 the IP address of discovery request 410 is entered into memory 414 in order to initiate a corresponding discovery procedure by discovery program 416. If the contrary is the case discovery request 410 is refused. This way unnecessary discovery procedures are avoided when computer peripheral 402 is frequently removed and reconnected to the computer network. For example the predetermined amount of time can be set to 6, 12 or 24 hours.

[0049] As an alternative to time stamping discovery request 410 by time stamp program component 422, the reception time of discovery request 410 can be used as a time reference. In this instance time stamp program component 422 is not required. This minimizes the complexity of the DHCP Server 400.

[0050] Again, it is to be noted that network server 400 and discovery server 412 are implemented on different server computers; however, network server 400 and discovery server 412 can also be implemented on the same server computer which is partitioned correspondingly.

[0051] Figure 5 is a corresponding flow chart. In step 500 an access request 406 from peripheral device 402 (e.g. a printer) is received by a network server 400 from a network node 402 which has been physically reconnected to the network. In step 502 the access request is processed by the network server 400. After successful completion of the processing in step 502, i.e. when the network node 402 is logically reconnected to the network, a discovery request is generated in step 504 and is time stamped by component 422 with a time stamp i.

[0052] In step 506 the time stamped discovery request is sent to the discovery server. In ~~response~~ response, the discovery server 412 performs a discovery procedure and enters the node ID of the network node for which the discovery procedure has been performed together with time stamp i in a discovery log file; step 507.

[0053] In step 508 the network node 402 is removed from the network. In step 510 the network node 402 is reconnected to the network. As a consequence, ~~consequence~~ a renewed access request from the reconnected network node 402 is received by the network server 412 in step ~~512 which~~ 512. The renewed access request is processed in step 514 and time stamped with time stamp j in step ~~516~~ 516, before the time stamp discovery request is sent to the discovery server in step 517.

[0054] When the renewed discovery request is received by the discovery server 414 the discovery server checks if time stamp j minus time stamp i is above a predefined threshold. In other ~~words~~ words, the discovery server 414 checks if a sufficient amount of time has lapsed between the first discovery request for which a discovery procedure has been performed in step 507 and the subsequent discovery request with time stamp j. This check is performed in step 518.

[0055] If the amount of time which has lapsed between the two subsequent discovery requests is below the threshold the control goes to step 519 where the renewed discovery request is rejected. If the contrary is the case a renewed discovery procedure is carried out in step 520.